

Prediction by a Hybrid of Wavelet Transform and Long-Short-Term-Memory Neural Network

Putu Sugiartawan, Reza Pulungan, and Anny Kartika Sari

Department of Computer Science and Electronics

Faculty of Mathematics and Natural Sciences

Universitas Gadjah Mada, Yogyakarta, Indonesia

Abstract—Data originating from some specific fields, for instance tourist arrivals, may exhibit a high degree of fluctuations as well as non-linear characteristics due to time varying behaviors. This paper proposes a new hybrid method to perform prediction for such data. The proposed hybrid model of wavelet transform and long-short-term memory (LSTM) recurrent neural network (RNN) is able to capture non-linear attributes in tourist arrival time series. Firstly, data is decomposed into constitutive series through wavelet transform. The decomposition is expressed as a function of a combination of wavelet coefficients, which have different levels of resolution. Then, LSTM neural network is used to train and simulate the value at each level to find the bias vectors and weighting coefficients for the prediction value. A sliding windows model is employed to capture the time series nature of the data. An evaluation is conducted to compare the proposed model with other RNN algorithms, *i.e.*, Elman RNN and Jordan RNN, as well as the combination of wavelet transform with each of them. The result shows that the proposed model has better performance in terms of training time than the original LSTM RNN, while the accuracy is better than the hybrid of wavelet-Elman and the hybrid of wavelet-Jordan.

Keywords—Wavelet Transform; Long-Short-Term Memory; Recurrent Neural Network; Time Series Prediction

I. INTRODUCTION

The growth in the number of visitors and tourism investments makes tourism become a key factor in export earnings, job creation, business development and infrastructure. Tourism has shifted and become one of the largest fast growing economic sectors in the world. Despite the global crises that occur several times, the number of international tourist trips continues to show positive growth. As shown by the data from BPS, the Indonesian Central Agency for Statistics, the number of tourists visiting Indonesia has increased from year to year. Travel and tourism directly contributes 2.1 trillion dollars to global GDP. It is more than doubled, compared to the automotive industry, and nearly 40 percent larger than the global chemical industry [1]. Travel and tourism sector is worth three quarters of the education sector, the banking sector, the mining sector, and the communications services sector. By knowing the number of the visitors to a country, the income of the country from the tourism sector can be predicted.

Fluctuation in the number of tourists visiting Indonesia in every year is not easy to predict. This has become a major problem for some parties such as hotels, restaurants and travel agents. This also causes those parties not able to devise good plannings for their business. The difficulty in determining the data traffic patterns is due to the existence of noise.

To overcome this problem, a technique is needed to separate the low frequency pattern from the high frequency pattern through the process of translation (shifting) and dilation (scaling) [2]. Wavelet transform can reveal aspects of frequencies in the frequency decomposition process [2], [3]. In [4], the use of wavelet sequence prediction models improves the effectiveness of multi layer perceptron (MLP) neural network. A merger between wavelet model and Kalman filter produces a powerful model for estimation technique [5], so does the merging of wavelet model with spectral analysis [6]. Evaluations conducted to several wavelet-RNN models show that the combination between wavelet and RNN usually produces smaller error value [7].

Recurrent neural networks have the capability to dynamically incorporate past experience due to internal recurrence [2]. RNNs can project the dynamic properties of the system automatically, so they are computationally more powerful than feed-forward networks, and the valuable approximation results are obtained for chaotic time series prediction [8], [9]. One of RNN models is long-short-term memory. This model works when there is a long delay, and is able to handle signals that have a mixture of low and high frequency components. The learning process of RNN models however requires a relatively long time because there is a context layer in the network architecture [10].

LSTM is a successful RNN architecture model to fix the vanishing gradient problem in neural network [11]. Sequence-based prediction of protein localization produces high prediction accuracy with LSTM and bidirectional model [12]. Comparing LSTM RNN model with random walk (RW), support-vector machine (SVM), single-layer feed forward (FFNN) and stacked autoencoder (SAE) shows that the LSTM RNN model produces higher accuracy and generalizes well [13]. Prediction of time series data for securities in Shanghai *ETF180* obtains a good accuracy. Using LSTM RNN, the result increases by 4 percents compared to the previous model, while data normalization can also improve accuracy [14].

In this paper a new hybrid algorithm for prediction, which is based on a combination of wavelet analysis and LSTM neural networks, is proposed. The proposed method is then applied for predicting tourist arrivals. The wavelet is employed to denoise the original signal and decompose the historical number of tourist arrivals into better series pattern for prediction. An LSTM neural network is used as a non-linear pattern recognition to estimate the training data signal and to compensate the error of wavelet-LSTM prediction. The proposed method is applied to tourist arrival data, which is

a set of 240 vector data.

The rest of the paper is organized as follows: The principles of the proposed method are described in Section II. Simulation result and the comparison of the proposed model with Elman and Jordan recurrent neural networks are presented in Section III. Finally, we conclude the paper and present future work in Section IV.

II. METHODOLOGY

Improving the accuracy of prediction can be performed by combining several different methods [15]. In this paper, LSTM neural network model is used to identify data pattern, while the wavelet method is employed to decompose input data. Prediction model using the hybrid of wavelet transform and LSTM neural network consists of the following phases:

- Phase 1: normalizing the data to values ranging between 0 and 1,
- Phase 2: decomposing data into constitutive series through wavelet transform,
- Phase 3: applying sliding windows to the data to form several variables, and
- Phase 4: recognizing data pattern using LSTM neural network model through data training and data testing.

The result of the proposed model is then compared with Elman RNN, Jordan RNN and LSTM RNN, as well as the hybrid of wavelet-Elman and the hybrid of wavelet-Jordan. The following subsections further elaborate the general design of the proposed model, the details of the wavelet transform as well as LSTM, data normalization and the evaluation metrics used.

A. Design of the proposed model

Fig. 1 depicts the flowchart of the proposed model. The model is divided into two processes, *i.e.*, training and testing, each of which further contains several processes. The first process in data training is to normalize the data with Min-Max normalization. The second process is data decomposition using wavelet transform. The purpose of wavelet transform is to divide the data into high and low frequencies. The third process is segmentation of time series data input using sliding windows. The next process is training using LSTM RNN. The training process uses 90 percent of data. The result of training process is a weight value for each neuron in the neural network and error value, which are shown by MSE and RMSE. After the weight value for each neuron is obtained, the testing process is started. The testing process produces error value. To restore the decomposed values to the original values, reconstruction and denormalization processes are performed. The purpose of reconstruction is to restore the data from high and low frequencies, while denormalization aims at restoring the data to original values.

In neural networks, data training is performed to gain bias values and weightings for prediction approximation and detail coefficients [3]. Bias and weighting coefficients generated from learning data are used in the testing process. The process is conducted iteratively to generate the prediction coefficient

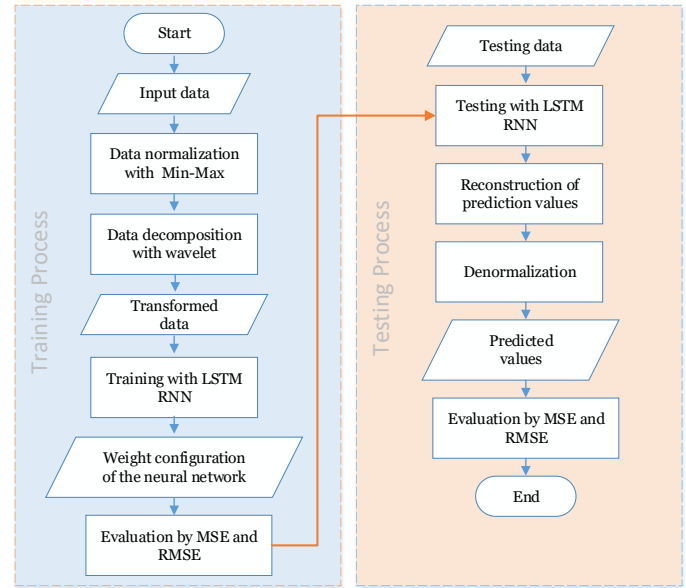


Fig. 1. Flowchart of the proposed model

details. Another learning process with the purpose of obtaining a prediction coefficient approximation has the same initial step. The learning does not use the input from the beginning, but is performed on approximation coefficients at the lowest levels (DWT 3). Once the simulation result has reached the desired target or maximum epoch, the learning finishes. The learning process uses back-propagation algorithm with the addition of a context layer [16] to accelerate the convergence towards the desired minimum value error.

B. Data normalization

Prior to the decomposition process, research data obtained from BPS must be normalized. The technique used provides linear transformation on original range of data, and is called Min-Mix normalization [17], [18]. The technique keeps the relationship among original data. Min-Max normalization is a simple technique, where the technique can specifically fit the data in a pre-defined boundary. A normalized value \hat{x} of a data point x_i in a predefined boundary $[C, D]$ is defined by:

$$\hat{x} = \frac{(x_i - x_{min})}{(x_{max} - x_{min})}(D - C) + C, \quad (1)$$

where x_{min} is the smallest value of the data, x_{max} is biggest value of the data, and $[x_{min}, x_{max}]$ is the range of the original data. The normalization process will produce a value ranging between 0 and 1. The biggest value will produce a value of 1, while the smallest value will produce a value of 0 in the normalization process.

To restore the normalization value to the original value, denormalization process is conducted. This aims at restoring the output of the value to be in the range beforehand. Given a normalized value \hat{x} , its denormalization, namely, the original data point x_i can be calculated by:

$$x_i = \frac{\hat{x}(x_{max} - x_{min})}{(D - C) + C} + x_{min}. \quad (2)$$

C. Data decomposition with wavelet transform

A wave is a function that moves up and down space at a time on a periodic basis, while wavelet is a restricted or localized wave [19]. Wavelet can also be regarded as a short-wave. The model provides a depiction of the frequency of the signal timing.

A wavelet transform (WT) is a time-frequency decomposition that provides a useful basis of time series in both time and frequency [8], [20], when the time series, like tourist arrival series, is non-stationary. Mother wavelet is the basic function used in wavelet transform [2] as it produces all functions used in the wavelet transformation through translation and scaling. The mother wavelet will determine the characteristics of the produced wavelet transform. Therefore, selection of the mother wavelet type must be done carefully in order to perform the transformation efficiently.

The wavelet transform can identify and analyze the signal moves. The purpose of analyzing the signal moves is to obtain information and the frequency spectrum at the same time. Discrete wavelet transform (DWT) is one of the wavelet transform development series. DWT works on two collections of functions called scaling functions and wavelet functions that are each associated with a low-pass filter and a high-pass filter [20]. The decomposition structure of wavelet transforms for level 3 is shown in Fig. 2.

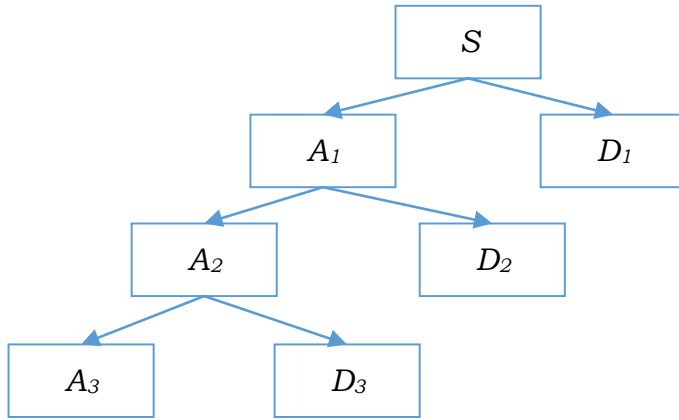


Fig. 2. Wavelet decomposition tree at level 3

The type of wavelet used in this research is *Haar* wavelet transform. It is a simple Daubechies wavelet, which is suitable to detect time-localized information and increases the performance of the prediction technique [21]. The Haar wavelet has two functions called approximation function and difference function. The approximation function produces a sequence of the averages between two consecutive data in the data input, while the difference function produces the current approximation sequence. Both functions are executed recursively and the process will stop when the element in the difference sequence is one [22]. The i -th approximation sequence (A_i) is given by:

$$A_i = \frac{A_{i-1}(1) + A_{i-1}(2)}{2} + \frac{A_{i-1}(3) + A_{i-1}(4)}{2} + \dots + \frac{A_{i-1}(n-1) + A_{i-1}(n)}{2}, \quad (3)$$

where $A_{i-1}(j)$ is the j -th element in the sequence (A_{i-1}) for $j = 1, 2, \dots, n$.

Decomposition process can be through one or more levels. Discrete wavelet series contain approximated series (A_t) and detail series (D_t). Dimensional signal can be divided into two parts, the high frequency and the low frequency parts. The high frequencies is analyzed with low-pass filter, while the low frequencies is analyzed with high-pass filter. Both frequency filters are used to analyze the different resolutions of the signal. The signal could be subsampled by 2, by discarding every second sample. The decomposition for each layers are represented by [8]:

$$y_{low}(k) = \sum_i x(i)h(2k-i), \text{ and} \quad (4)$$

$$y_{high}(k) = \sum_i x(i)g(2k-i), \quad (5)$$

where y_{low} and y_{high} is a low-pass and high-pass filters, respectively, both subsampling by 2 [8]. k refers to the time decomposition and the original signal data $x(i)$ is passed through to a high-pass filter $g(\cdot)$ and a low-pass filter $h(\cdot)$. The above functions can be reused in the next decomposition. DWT coefficient consists of the output of high-pass and low-pass filters.

The high and low-pass filter functions are followed in reverse order by the reconstruction. The signal at each layer is upsampled by two, through the synthesis filter $g'(\cdot)$ and $h'(\cdot)$ and added to each other [8]. The reconstruction for each layer is given by:

$$x(i) = \sum_k y_{high}(k)g'(-n+2k) + y_{low}(k)h'(-n+2k). \quad (6)$$

The reconstruction process aims at returning the original values of data. Reconstruction is started by combining DWT coefficients which are at the end of the previous decomposition upsampled by 2 ($\uparrow 2$) through a high-pass filter and low-pass filter. The reconstruction process is completely the opposite of the decomposition process according to the level of decomposition [19].

D. Sliding windows technique

Sliding windows technique is a kind of processing method of concept drift in data streams, which has many applications in intrusion detection. The essence of this technique is data update mechanism. The data stream (x) is divided into several parts of data blocks. When sliding window moves to the next block, new block is added to the window at intervals, and the oldest block is deleted. Through this dynamic sample selection method, the sample for modeling is updated [23].

The technique of sliding windows comes with a particular size of window; and this impacts the size of sample data. Suppose that $\langle x_0, x_1, x_2, \dots, x_{n-1}, x_n, x_{n+1}, \dots \rangle$ is a series of time-series data. When the window size is fixed at k , the data interval will be changed to $\langle x_{i-k}, x_{i-k+1}, \dots, x_i, x_{i+1} \rangle$ and has different data streams from older data streams. As shown in Fig. 3, the data interval is $\langle x_{i-2}, x_{i-1}, x_i, x_{i+1} \rangle$ for window size of 3, where the value of x_{i+1} is obtained from $\{x_{i-2}, x_{i-1}, x_i\}$ values.

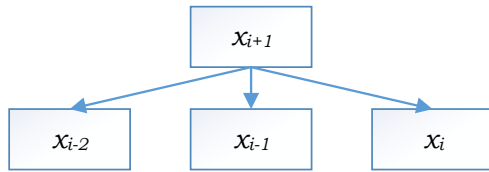


Fig. 3. Sliding windows technique with window size of 3

In the proposed model, the window size in the sliding windows depends on the number of data. The window size determines prediction accuracy. Based on several experiments, the window size selected in this paper is 3, because it provides the best value and minimizes data reduction. The number of data is 120 after decomposition process with wavelet. Then, the sliding windows process generates 3 data input variables, 1 output variable, and there will be 117 data. The number of data is reduced because the windows eliminates the last data record.

The implementation of sliding windows technique on RNN is shown in Fig. 4. The number of previous values used as input $\{\dots, x_{n-2}, x_{n-1}\}$ depends on the window size k , while the current value becomes the target y value at the output layer. Since the window size is 3, $\{x_{n-2}, x_{n-1}, x_n\}$ are the input values, while x_{n+1} is the output value of the RNN architecture.

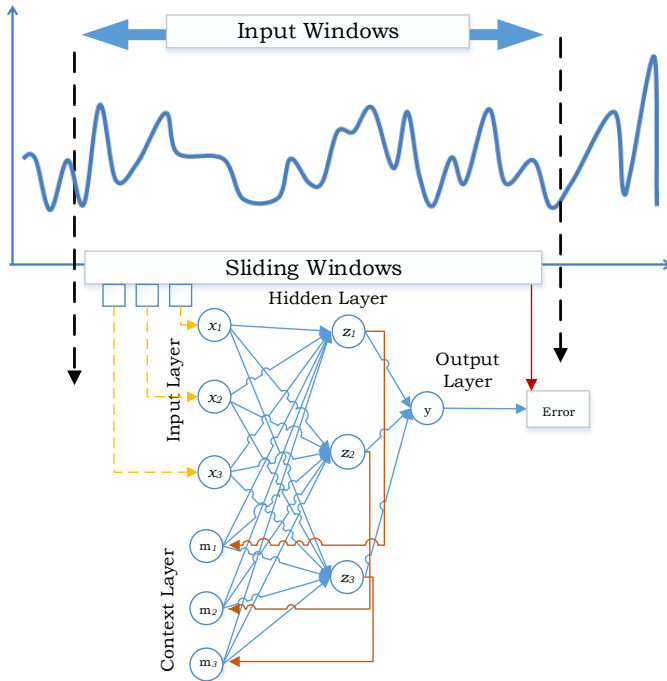


Fig. 4. Illustration of sliding windows technique on RNN

E. Data training with LSTM RNN model

Fig. 5 depicts the general architecture of an LSTM RNN [9]. An LSTM RNN is an artificial neural network structure for the solution of vanishing gradient. The algorithm works when there is a long delay, and can handle signals that have a mixture of low and high frequency components. The LSTM contains special units called memory blocks in the recurrent hidden layer as shown in Fig. 5. The memory blocks

contain memory cells with self-connections, storing the temporal state of the network in addition to special multiplicative units called gates to control the flow of information.

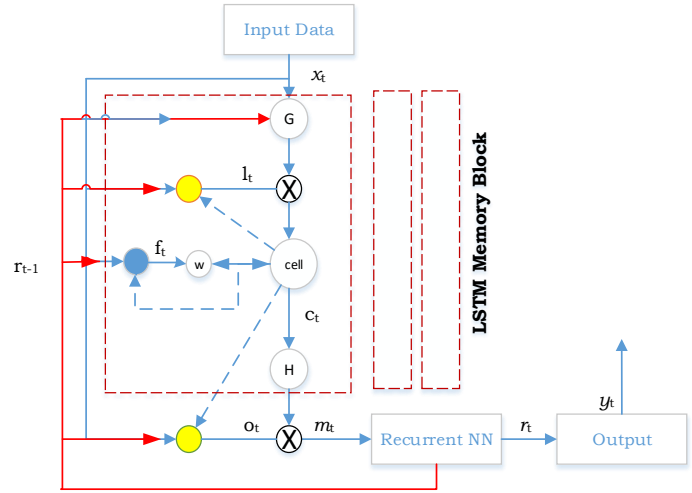


Fig. 5. The general LSTM-RNN architecture

Each memory block in the original architecture contains an input gate (G) and an output gate (H). The input gate (G) controls the flow of input activation (X_t) into the memory cell. The output flow (C_i) of cell activation is controlled by the output gate into the rest of the network; the next process is memory block added by forget gate (w) [24].

The forget gate (w) scales the internal state of the cell before adding it as input to the cell through the self-recurrent connection of the cell, therefore adaptively forgetting or resetting the cell's memory. In addition, modern LSTM architectures contain peephole connections from its internal cells to the gates in the same cell to learn the precise timing of the outputs [25].

An LSTM network computes from an input sequence $x = \langle x_1, \dots, x_T \rangle$ an output sequence $y = \langle y_1, \dots, y_T \rangle$ by calculating the network unit activations using the following equations, for $t = 1, \dots, T$ [25]:

$$i_t = \sigma(W_{ix}x_t + W_{im}m_{t-1} + W_{ic}c_{t-1} + b_i), \quad (7)$$

$$f_t = \sigma(W_{fx}x_t + W_{fm}m_{t-1} + W_{fc}c_{t-1} + b_f), \quad (8)$$

$$c_t = f_t \odot c_{t-1} + i_t \odot g(W_{cx}x_t + W_{cm}m_{t-1} + b_c), \quad (9)$$

$$o_t = \sigma(W_{ox}x_t + W_{om}m_{t-1} + W_{oc}c_t + b_o), \quad (10)$$

$$m_t = o_t \odot h(c_t), \quad (11)$$

$$y_t = \phi(W_{ym}m_t + b_y). \quad (12)$$

We use W to denote the weight matrices (e.g., W_{ix} is the matrix of weights from the input gate to the input), W_{ic} , W_{fc} , and W_{oc} are diagonal weight matrices for peephole connections. Vector b denotes a bias vector and b_i is the input gate bias vector. Function σ is the logistic Sigmoid function, i is the input gate, f is the forget gate, o is the output gate, and c denotes the cell activation vector. Functions i , f , o and c have the same size as the cell output activation vector m . Operator \odot is the element-wise product of the vectors. Function g is the cell input function, while function h is a cell output activation

function. Activation functions used in this paper are tanh and \emptyset , which denotes the network output activation function.

Fig. 6 shows the architecture of the LSTM-RNN in the proposed model. The number of hidden layer in the model is 1, hence the context layer must be 1. The training process is performed iteratively until the output value approximates the original value. The number of epoch in the training process is 10^5 with the learning rate of 0.1. $\{x_1, x_2, x_3\}$ is an input data at input layer, $\{z_1, z_2, z_3\}$ is a hidden layer unit, $\{m_1, m_2, m_3\}$ is a context layer unit, and y is an output value (the result of the RNN architecture).

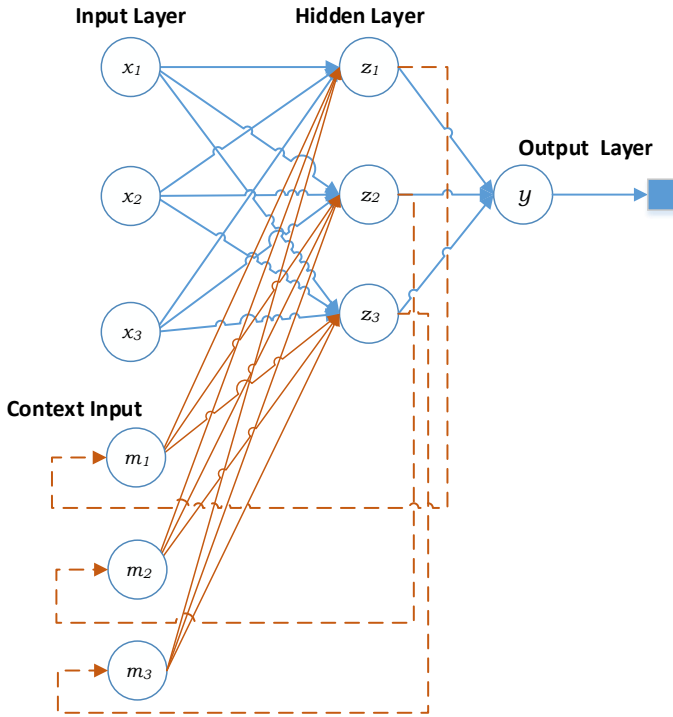


Fig. 6. The proposed LSTM-RNN architecture

F. Prediction accuracy

To measure and evaluate the prediction accuracy of the proposed hybrid model, mean square error (MSE) and root mean square error (RMSE) methods are used. Let y_i be the measured value of time i or the targeted minimum error on a neural network, f_i is the predicted value at time i obtained from a particular model M , and n be the number of sample data. Then, MSE is given by [26]:

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - f_i), \quad (13)$$

and RMSE is given by:

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - f_i)}. \quad (14)$$

RMSE is used because it has a good performance to distribution error and could satisfy the triangle inequality requirement for a distance metric [27]. MSE is an error

function for evaluating the performance and efficiency of the forecasting methods. MSE can compare point-by-point for overall performance measure method of the actual time series values and the forecast value [28]. If the values of RMSE and MSE are lower, the accuracy of the model is better.

III. RESULT AND ANALYSIS

The proposed model is applied to predict tourist arrivals in Indonesia. Data used in this research consists of the number of tourist visits to Indonesia from 1995 to 2014 in each month. Hence, there is 240 data records in total. Ninety percent of the data is for training, while 10 percent is for testing. The model is then compared to the original LSTM RNN and other types of RNN, *i.e.*, Elman RNN and Jordan RNN, as well as the hybrid of wavelet-Elman and the hybrid of wavelet-Jordan. These models will be compared in terms of accuracy and the time required for data training and data testing.

The first process is data normalization which aims to simplify calculations, reduce the value range, and make the training process faster. The normalization, which utilizes Min-Max technique, produces data in the range of $[0, \dots, 1]$. Data is then transformed using Haar wavelet function, and 3 levels of decomposition is applied. The wavelet transform is implemented in Matlab.

Fig. 7 shows three levels of wavelet decomposition of tourist arrival time-series data. Graph S shows the time series data after normalization by Min-Max technique. The first level of decomposition process using Haar function produces high and low frequencies, as shown in graph A_1 and D_1 . Data in each decomposition is divided into two, which each at level 3 produces 30 data records. The decomposition with level three A -series data (graph A_3) has the lowest frequency content and this tends to be incompatible with prediction.

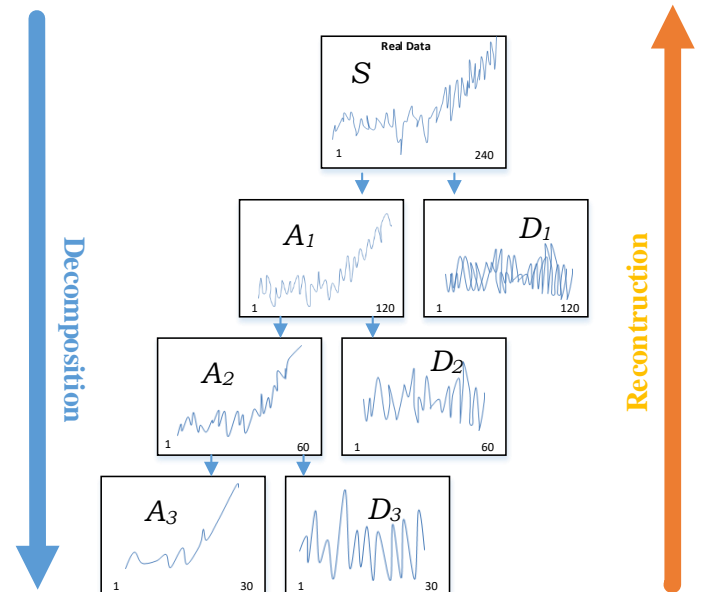


Fig. 7. Three-level wavelet decomposition of time series

The prediction is performed to data in each of graphs S , A_1 , A_2 and A_3 . Here, LSTM neural network with sliding

windows technique with window size 3 is employed. The LSTM model includes 3 input data, a hidden layer with 4 LSTM blocks or neurons and an output layer that produces a single value prediction. The activation function for LSTM blocks is binary Sigmoid. The network is trained for 105 epochs and a batch size of 1 is used. Table I depicts LSTM parameters for data training. The prediction is implemented using Theano library provided in Python.

TABLE I. LSTM PARAMETERS FOR DATA TRAINING

Parameter	Value	Information
Error Target (MSE)	0.001	Prediction error value
Learning Rate	0.01	Pace of learning
Hidden Layer	1	Number of hidden layers
Input Layer	3	Number of input layers
Output Layer	1	Number of output layers
Epoch Max	10^5	Maximum epoch
Transfer Function	Binary Sigmoid	
Weight	Random (0, 1)	

The proposed model is compared to hybrid wavelet-Elman, hybrid wavelet-Jordan, the original Elman RNN, Jordan RNN, and LSTM RNN. Tabel II depicts the results of the comparison between the proposed model and other models. The comparison between the models is based on the accuracy and the time required for data training and data testing. The result shows that the hybrid of wavelet transform and LSTM has better performance than other hybrid models in terms of accuracy. Nevertheless, the proposed model takes the most time for training process compared to other hybrid models. On the other hand, the hybrid model is able to reduce training time on the original LSTM model by 28 minutes. All hybrid models are able to reduce training time significantly; an average of twice faster than the training time required without the use of wavelet transform. Hybrid wavelet-Jordan produces the smallest training time among the models, but the error value is quite high compared to the original LSTM and the hybrid wavelet-LSTM model.

TABLE II. COMPARISON OF PERFORMANCE USING VARIOUS PREDICTION METHODS

Algorithm	MSE	RMSE	Training Time
Hybrid Wavelet and LSTM (proposed)	0.11853	0.34428	00:31:58
Hybrid Wavelet and Elman	0.14521	0.38107	00:12:37
Hybrid Wavelet and Jordan	0.21776	0.46664	00:09:52
LSTM RNN	0.00464	0.21541	00:59:11
Elman RNN	0.24791	0.49796	00:20:16
Jordan RNN	0.35256	0.59377	00:17:31

Fig. 8 depicts the comparison of the real data with the decomposition of signal produced using wavelet transform as well as the prediction of signal using LSTM neural network. In prediction where all data is used, the result is very similar to the original data. Due to decomposition, data is then reduced in each level of decomposition. Using data produced by DWT level 1, prediction result is still recognizable, and the accuracy is not too bad. However, using data of DWT levels 2 and 3, prediction results become more unrecognizable; hence, there is a big gap between real data and predicted data. This means that the deeper the decomposition process, the more predicted data will be unrecognizable.

Table III compares the MSE of training and testing in each decomposition level to show the influence of data reduction, as



Fig. 8. Data training and data testing using different levels of decomposition

the result of decomposition, to the accuracy. Prediction using the original LSTM RNN model without wavelet decomposition (Data S) generates the smallest error value compared to prediction using wavelet decomposition. This means that as the training data reduces as the result of the decomposition process, more data cannot be identified at the time of testing. The hybrid model at level 1 (A_1) generates an error value that is smaller than the hybrid model at level 2 (A_2) and 3 (A_3).

TABLE III. COMPARISON OF ACCURACY WITH VARIOUS LEVELS OF DATA DECOMPOSITION WITH MSE

Data	Training's MSE	Training Time	Testing's MSE
S	0.00464	00:59:11	0.02406
DWT level 1 (A_1)	0.11853	00:32:28	0.12870
DWT level 2 (A_2)	0.23689	00:14:16	0.02762
DWT level 3 (A_3)	0.44674	00:05:50	1.15110

From the experiments we have conducted, it can be inferred that the advantage of the use of the proposed model is mainly for shortening the time for data training rather than for increasing prediction accuracy. The lower accuracy compared to the original LSTM is due to the data reduction as the result of decomposition in wavelet transform. Hence, the proposed model can reduce the time of training process but not the error values.

IV. CONCLUSION AND FUTURE WORK

In this paper, a hybrid model of wavelet transform and LSTM neural network is proposed to predict the number of tourist arrivals in Indonesia. This model incorporates wavelet and LSTM neural network to predict the number of tourist arrivals each month. The wavelet algorithm is used to decompose time series data into the data of low frequency and high frequency, which is proven to reduce the time for data training. The LSTM neural network is employed for training and testing the results of wavelet transform. The predicted outcome of the proposed hybrid model is compared to the original LSTM,

Elman and Jordan RNN, as well as the hybrid of wavelet-Elman and the hybrid of wavelet-Jordan. The evaluation shows that the hybrid model of wavelet and LSTM method gives better training time than the original LSTM, Elman, and Jordan RNNs. Furthermore, this method is able to predict the number of tourist arrival more accurately than other hybrid methods.

One of the issues which is interesting for future work is to employ clustering method, such as k -means, to form the hybrid of LSTM RNN and clustering for time series prediction. The purpose of this is to compare the training time and the accuracy, to know which hybrid model can give better results.

REFERENCES

- [1] The World Tourism Organization UNWTO, "International tourist arrivals up 4% in the first four months of 2015," *Press Release no. PR15048*, 2015. [Online]. Available: <http://media.unwto.org/press-release/2015-07-08/international-tourist-arrivals-4-first-four-months-2015>
- [2] F. Murtagh, J.-L. Starck, and O. Renaud, "On neuro-wavelet modeling," *Decision Support System*, vol. 37, no. 4, pp. 475–484, September 2004.
- [3] U. Lotric, "Wavelet based denoising integrated into multilayered perceptron," *Neurocomputing*, vol. 62, pp. 179–196, December 2004.
- [4] L. Wang, K. K. Teo, and Z. Lin, "Predicting time series with wavelet packet neural networks," in *Neural Networks, 2001. Proceedings. IJCNN '01. International Joint Conference on*, vol. 3, School of Electrical and Electronic Engineering nanyang Technological University, nanyang Avenue Singapore 639798: IEEE, 2001, pp. 1593–1597.
- [5] X. Zhao, J. Lu, W. P. A. Putranto, and T. Yahagi, "Nonlinear time series prediction using wavelet networks with Kalman filter based algorithm," in *2005 IEEE International Conference on Industrial Technology*. IEEE, December 2005, pp. 1226–1230.
- [6] G. Dominguez, M. Guevara, M. Mendoza, and J. Zamora, "A wavelet-based method for time series forecasting," in *2012 31st International Conference of the Chilean Computer Science Society*. IEEE, November 2012, pp. 91–94.
- [7] L. F. Ortega, "A neuro-wavelet method for the forecasting of financial time series," in *International Conference on Soft Computing and Applications 2012*, vol. 1, WCECS. San Francisco, USA: International Association of Engineers, October 2012, pp. 24–26.
- [8] N. Terzija, "Robust digital image watermarking algorithms for copyright protection," Ph.D. dissertation, Universität Duisburg-Essen, January 2006.
- [9] H. Sak, A. W. Senior, and F. Beaufays, "Long short-term memory recurrent neural network architectures for large scale acoustic modeling," in *INTERSPEECH 2014: 15th Annual Conference of the International Speech Communication Association*, 2014, pp. 338–342.
- [10] P. Fryzlewicz, S. V. Belleghem, and R. von Sachs, "Forecasting non-stationary time series by wavelet process modelling," *Annals of the Institute of Statistical Mathematics*, vol. 55, no. 4, pp. 737–764, December 2003.
- [11] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997. [Online]. Available: <http://dx.doi.org/10.1162/neco.1997.9.8.1735>
- [12] T. Thireou and M. Reczko, "Bidirectional long short-term memory networks for predicting the subcellular localization of eukaryotic proteins," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 4, no. 3, pp. 441–446, July 2007.
- [13] Y. Tian and L. Pan, "Predicting short-term traffic flow by long short-term memory recurrent neural network," in *2015 IEEE International Conference on Smart City/SocialCom/SustainCom (SmartCity)*, December 2015, pp. 153–158.
- [14] K. Chen, Y. Zhou, and F. Dai, "A LSTM-based method for stock returns prediction: A case study of China stock market," in *2015 IEEE International Conference on Big Data (Big Data)*, October 2015, pp. 2823–2824.
- [15] M. Shafie-khah, M. P. Moghaddam, and M. Sheikh-El-Eslami, "Price forecasting of day-ahead electricity markets using a hybrid forecast method," *Energy Conversion and Management*, vol. 52, no. 5, pp. 2165–2169, May 2011.
- [16] Z. Zainuddin, N. Mahat, and Y. A. Hassan, "Improving the convergence of the backpropagation algorithm using local adaptive techniques," *International Journal of Computer, Electrical, Automation, Control and Information Engineering*, vol. 1, no. 1, pp. 184–187, December 2007.
- [17] S. K. Panda and P. K. Jana, "Efficient task scheduling algorithms for heterogeneous multi-cloud environment," *The Journal of Supercomputing*, vol. 71, no. 4, pp. 1505–1533, January 2015.
- [18] —, "A multi-objective task scheduling algorithm for heterogeneous multi-cloud environment," in *2015 International Conference on Electronic Design, Computer Networks Automated Verification (EDCAV)*. IEEE, January 2015, pp. 82–87.
- [19] D. Sripathi, "Efficient implementations of discrete wavelet transforms using fpgas," Master's thesis, Florida State University, November 2003.
- [20] T. W. Joo and S. B. Kim, "Time series forecasting based on wavelet filtering," *Expert Systems with Applications*, vol. 42, no. 8, pp. 3868–3874, May 2015.
- [21] C. Stolojescu, I. Railean, S. Moga, P. Lenca, and A. Isar, "A wavelet based prediction method for time series," in *Proceedings of Stochastic Modeling Techniques and Data Analysis (SMTDA2010) International Conference*, Chania, Greece, 2010.
- [22] K. Kawagoe and T. Ueda, "A similarity search method of time series data with combination of Fourier and wavelet transforms," in *Proceedings Ninth International Symposium on Temporal Representation and Reasoning*, 2002, pp. 86–92.
- [23] J. Shi and L. Cheng, "Financial crisis dynamic prediction based on sliding window technology and Mahalanobis-Taguchi system," in *2011 International Conference of Information Technology, Computer Engineering and Management Sciences*, vol. 4, Sept 2011, pp. 65–68.
- [24] F. A. Gers, J. Schmidhuber, and F. Cummins, "Learning to forget: Continual prediction with LSTM," *Neural Computing*, vol. 12, no. 10, pp. 2451–2471, October 2000.
- [25] F. A. Gers, N. N. Schraudolph, and J. Schmidhuber, "Learning precise timing with LSTM recurrent networks," *Journal of Machine Learning Research*, vol. 3, pp. 115–143, 2002.
- [26] B. Rahmadi and Supriyadi, "Early model of traffic sign reminder based on neural network," *TELKOMNIKA*, vol. 10, no. 4, pp. 749–758, December 2012.
- [27] T. Chai and R. R. Draxler, "Root mean square error (RMSE) or mean absolute error (MAE)? - Arguments against avoiding RMSE in the literature," *Geoscientific Model Development*, vol. 7, pp. 1247–1250, Jun. 2014.
- [28] M. Yousefi, D. Hooshyar, M. Yousefi, W. Khaksar, K. S. M. Sahari, and F. B. I. Alnaimi, "An artificial neural network hybrid with wavelet transform for short-term wind speed forecasting: A preliminary case study," in *2015 International Conference on Science in Information Technology (ICSITech)*. IEEE, October 2015, pp. 95–99.